

# **LAPORAN UAS**

## **Teknologi Data**

**UAS**



**Oleh:**

**Bagus Satria putra**

**1841720146 TI-3F**

**PROGRAM STUDI D-IV TEKNIK  
INFORMATIKA JURUSAN TEKNOLOGI  
INFORMASI POLITEKNIK NEGERI  
MALANG  
26 JUNI 2021**

No.	Keterangan
	<p>Membuat program naivebayes.py</p> <pre> import pandas as pd from probaspek import ProbAspek  class NaiveBayes:      def __init__(self):         pass          # TODO: [LANGKAH-2] Buat property untuk menampung data dari file CSV         self.data_training = None          # TODO: [Langkah- 3] Buat variabel dictionary untuk menampung matriks Probabilitas untuk semua aspek          self.aspek_umur = {'Tua': None, 'Muda': None}         self.aspek_status = {'Belum Kawin': None, 'Kawin': None, 'Cerai': None}         self.aspek_pendidikan = {'Tidak Sekolah': None, 'SD': None, 'SLTP': None, 'SLTA': None, 'Sarjana': None}         self.aspek_tanggungan = {'0': None, '1': None, '2': None, '3': None, '4': None, '5': None}         self.aspek_pekerjaan = {'Tiada': None, 'Buruh Lepas': None, 'Petani': Non e, 'Aparatur Negara': None}         self.aspek_penghasilan = {'Tiada': None, 'Rendah': None, 'Sedang': None, 'Tinggi': None}          # TODO: [Langkah-4] Buat variabel untuk menampung Prior Probability         self.prior_probability = {'Miskin': 0, 'Tidak Miskin': 0}          # TODO: [LANGKAH-5] Load data training dari file CSV         def load_data_training(self):             self.data_training = pd.read_csv('daftar_masyarakat_miskin.csv', sep=';')             print(self.data_training)             print('-----')          # TODO: [LANGKAH- 6] Membuat object ProbAspek untuk semua nilai pada aspek, sekaligus menghitung ju mlah miskin dan tidak miskin         def buat_prob_aspek(self, nama_aspek: str, nilai_aspek: str) -&gt; ProbAspek:             #probabilitas aspek kosong             prob_aspek = ProbAspek(nama_aspek, nilai_aspek)             #menghitung jumlah miskin dan tidak miskin </pre>

```

        tua_miskin = self.data_training.loc[(self.data_training[nama_aspek] == nilai_aspek) &
                                            (self.data_training['Actual Class'] == 'Miskin')]
        tua_tidak_miskin = self.data_training.loc[(self.data_training[nama_aspek] == nilai_aspek) &
                                                  (self.data_training['Actual Class'] == 'Tidak Miskin')]
        prob_aspek.jml_miskin = len(tua_miskin)
        prob_aspek.jml_tidak_miskin = len(tua_tidak_miskin)
        return prob_aspek

# TODO: [LANGKAH-7] Mengisi semua nilai pada matriks probabilitas aspek
def mulai_training(self):
    # Aspek Umur
    pu_tua = self.buat_prob_aspek('Umur', 'Tua')
    pu_muda = self.buat_prob_aspek('Umur', 'Muda')

    # Jadikan array
    arr_pu = [pu_tua, pu_muda]
    # Hitung total masing-masing nilai aspek berapa kali muncul di miskin dan tidak miskin
    total_u = ProbAspek.hitung_jml_total_aspek(arr_pu)

    # Hitung probabilitas aspek untuk masing-masing nilai aspek
    pu_tua.hitung_p_aspek_miskin(total_u['Miskin']).hitung_p_aspek_tidak_miskin(total_u['Tidak Miskin'])
    pu_muda.hitung_p_aspek_miskin(total_u['Miskin']).hitung_p_aspek_tidak_miskin(total_u['Tidak Miskin'])

    # Print matriks probabilitas, tetapi bentuknya vertikal, bukan tabel
    ProbAspek.print_matrix_probabilitas(arr_pu)
    self.aspek_umur['Tua'] = pu_tua
    self.aspek_umur['Muda'] = pu_muda

    # TODO :1.1 Status
    # Aspek Status
    ps_belum_kawin = self.buat_prob_aspek('Status', 'Belum Kawin')
    ps_kawin = self.buat_prob_aspek('Status', 'Kawin')
    ps_cerai = self.buat_prob_aspek('Status', 'Cerai')
    # Jadikan array
    arr_ps = [ps_belum_kawin, ps_kawin, ps_cerai]
    # Hitung total masing-masing nilai aspek berapa kali muncul di miskin dan tidak miskin

```

```

total_s = ProbAspek.hitung_jml_total_aspek(arr_ps)
# Hitung probabilitas aspek untuk masing-masing nilai aspek
ps_belum_kawin.hitung_p_aspek_miskin(total_s['Miskin']).hitung_p_aspek_tidak_miskin(total_s['Tidak Miskin'])
ps_kawin.hitung_p_aspek_miskin(total_s['Miskin']).hitung_p_aspek_tidak_miskin(total_s['Tidak Miskin'])
ps_cerai.hitung_p_aspek_miskin(total_s['Miskin']).hitung_p_aspek_tidak_miskin(total_s['Tidak Miskin'])

# Print matrix probabilitas, tetapi bentuknya vertikal, bukan tabel
ProbAspek.print_matrix_probabilitas(arr_ps)
self.aspek_status['Belum Kawin'] = ps_belum_kawin
self.aspek_status['Kawin'] = ps_kawin
self.aspek_status['Cerai'] = ps_cerai

# TODO :1.2 Pendidikan
# Aspek Pendidikan
pn_tidak_sekolah = self.buat_prob_aspek('Pendidikan', 'Tidak Sekolah')
pn_sd = self.buat_prob_aspek('Pendidikan', 'SD')
pn_sltp = self.buat_prob_aspek('Pendidikan', 'SLTP')
pn_slta = self.buat_prob_aspek('Pendidikan', 'SLTA')
pn_sarjana = self.buat_prob_aspek('Pendidikan', 'Sarjana')

# Jadikan array
arr_pn = [pn_tidak_sekolah, pn_sd, pn_sltp, pn_slta, pn_sarjana]
# Hitung total masing-masing nilai aspek berapa kali muncul di miskin dan tidak miskin
total_pn = ProbAspek.hitung_jml_total_aspek(arr_pn)
# Hitung probabilitas aspek untuk masing-masing nilai aspek
pn_tidak_sekolah.hitung_p_aspek_miskin(total_pn['Miskin']).hitung_p_aspek_tidak_miskin(total_pn['Tidak Miskin'])
pn_sd.hitung_p_aspek_miskin(total_pn['Miskin']).hitung_p_aspek_tidak_miskin(total_pn['Tidak Miskin'])
pn_sltp.hitung_p_aspek_miskin(total_pn['Miskin']).hitung_p_aspek_tidak_miskin(total_pn['Tidak Miskin'])
pn_slta.hitung_p_aspek_miskin(total_pn['Miskin']).hitung_p_aspek_tidak_miskin(total_pn['Tidak Miskin'])
pn_sarjana.hitung_p_aspek_miskin(total_pn['Miskin']).hitung_p_aspek_tidak_miskin(total_pn['Tidak Miskin'])

# Print matrix probabilitas, tetapi bentuknya vertikal, bukan tabel
ProbAspek.print_matrix_probabilitas(arr_pn)
self.aspek_pendidikan['Tidak Sekolah'] = pn_tidak_sekolah
self.aspek_pendidikan['SD'] = pn_sd
self.aspek_pendidikan['SLTP'] = pn_sltp

```

```

self.aspek_pendidikan['SLTA'] = pn_slta
self.aspek_pendidikan['Sarjana'] = pn_sarjana

# TODO :1.3 Tanggungan
# Aspek Tanggungan
pt_nol = self.buat_prob_aspek('Tanggungan', 'Tidak Ada')
pt_satu = self.buat_prob_aspek('Tanggungan', '1')
pt_dua = self.buat_prob_aspek('Tanggungan', '2')
pt_tiga = self.buat_prob_aspek('Tanggungan', '3')
pt_empat = self.buat_prob_aspek('Tanggungan', '4')
pt_lima = self.buat_prob_aspek('Tanggungan', '5')

# Jadikan array
arr_pt = [pt_nol, pt_satu, pt_dua, pt_tiga, pt_empat, pt_lima]
# Hitung total masing-
masing nilai aspek berapa kali muncul di miskin dan tidak miskin
total_pt = ProbAspek.hitung_jml_total_aspek(arr_pt)
# Hitung probabilitas aspek untuk masing-masing nilai aspek
pt_nol.hitung_p_aspek_miskin(total_pt['Miskin']).hitung_p_aspek_tidak_mis
kin(total_pt['Tidak Miskin'])
pt_satu.hitung_p_aspek_miskin(total_pt['Miskin']).hitung_p_aspek_tidak_mi
skin(total_pt['Tidak Miskin'])
pt_dua.hitung_p_aspek_miskin(total_pt['Miskin']).hitung_p_aspek_tidak_mis
kin(total_pt['Tidak Miskin'])
pt_tiga.hitung_p_aspek_miskin(total_pt['Miskin']).hitung_p_aspek_tidak_mi
skin(total_pt['Tidak Miskin'])
pt_empat.hitung_p_aspek_miskin(total_pt['Miskin']).hitung_p_aspek_tidak_m
iskin(total_pt['Tidak Miskin'])
pt_lima.hitung_p_aspek_miskin(total_pt['Miskin']).hitung_p_aspek_tidak_mi
skin(total_pt['Tidak Miskin'])

# Print matrix probabilitas, tetapi bentuknya vertikal, bukan tabel
ProbAspek.print_matrix_probabilitas(arr_pt)
self.aspek_tanggungan['Tidak Ada'] = pt_nol
self.aspek_tanggungan['1'] = pt_satu
self.aspek_tanggungan['2'] = pt_dua
self.aspek_tanggungan['3'] = pt_tiga
self.aspek_tanggungan['4'] = pt_empat
self.aspek_tanggungan['5'] = pt_lima

# TODO :1.4 Pekerjaan
# Aspek Pekerjaan
pk_tiada = self.buat_prob_aspek('Pekerjaan', 'Tiada')
pk_buruh = self.buat_prob_aspek('Pekerjaan', 'Buruh Lepas')
pk_petani = self.buat_prob_aspek('Pekerjaan', 'Petani')

```

```

pk_aparatur = self.buat_prob_aspek('Pekerjaan', 'Aparatur Negara')

# Jadikan array
arr_pk = [pk_tiada, pk_buruh, pk_petani, pk_aparatur]
# Hitung total masing-
masing nilai aspek berapa kali muncul di miskin dan tidak miskin
total_pk = ProbAspek.hitung_jml_total_aspek(arr_pk)
# Hitung probabilitas aspek untuk masing-masing nilai aspek
pk_tiada.hitung_p_aspek_miskin(total_pk['Miskin']).hitung_p_aspek_tidak_miskin(total_pk['Tidak Miskin'])
pk_buruh.hitung_p_aspek_miskin(total_pk['Miskin']).hitung_p_aspek_tidak_miskin(total_pk['Tidak Miskin'])
pk_petani.hitung_p_aspek_miskin(total_pk['Miskin']).hitung_p_aspek_tidak_miskin(total_pk['Tidak Miskin'])
pk_aparatur.hitung_p_aspek_miskin(total_pk['Miskin']).hitung_p_aspek_tidak_miskin(total_pk['Tidak Miskin'])

# Print matrix probabilitas, tetapi bentuknya vertikal, bukan tabel
ProbAspek.print_matrix_probabilitas(arr_pk)
self.aspek_pekerjaan['Tiada'] = pk_tiada
self.aspek_pekerjaan['Buruh Lepas'] = pk_buruh
self.aspek_pekerjaan['Petani'] = pk_petani
self.aspek_pekerjaan['Aparatur Negara'] = pk_aparatur

# TODO :1.5 Penghasilan
# Aspek Penghasilan
ph_tiada = self.buat_prob_aspek('Penghasilan', 'Tiada')
ph_rendah = self.buat_prob_aspek('Penghasilan', 'Rendah')
ph_sedang = self.buat_prob_aspek('Penghasilan', 'Sedang')
ph_tinggi = self.buat_prob_aspek('Penghasilan', 'Tinggi')

# Jadikan array
arr_ph = [ph_tiada, ph_rendah, ph_sedang, ph_tinggi]
# Hitung total masing-
masing nilai aspek berapa kali muncul di miskin dan tidak miskin
total_ph = ProbAspek.hitung_jml_total_aspek(arr_ph)
# Hitung probabilitas aspek untuk masing-masing nilai aspek
ph_tiada.hitung_p_aspek_miskin(total_ph['Miskin']).hitung_p_aspek_tidak_miskin(total_ph['Tidak Miskin'])
ph_rendah.hitung_p_aspek_miskin(total_ph['Miskin']).hitung_p_aspek_tidak_miskin(total_ph['Tidak Miskin'])
ph_sedang.hitung_p_aspek_miskin(total_ph['Miskin']).hitung_p_aspek_tidak_miskin(total_ph['Tidak Miskin'])

```

```

        ph_tinggi.hitung_p_aspek_miskin(total_ph['Miskin']).hitung_p_aspek_tidak_miskin(total_ph['Tidak Miskin'])

        # Print matrix probabilitas, tetapi bentuknya vertikal, bukan tabel
        ProbAspek.print_matrix_probabilitas(arr_ph)
        self.aspek_penghasilan['Tiada'] = ph_tiada
        self.aspek_penghasilan['Rendah'] = ph_rendah
        self.aspek_penghasilan['Sedang'] = ph_sedang
        self.aspek_penghasilan['Tinggi'] = ph_tinggi

        # TODO: [LANGKAH-8] Menghitung prior probability
        def hitung_prior_probability(self):
            pp_miskin = self.buat_prob_aspek('Actual Class', 'Miskin')
            pp_tidak_miskin = self.buat_prob_aspek('Actual Class', 'Tidak Miskin')
            arr_pp = (pp_miskin, pp_tidak_miskin)
            total_pp = ProbAspek.hitung_jml_total_aspek(arr_pp)
            self.prior_probability['Miskin'] = total_pp['Miskin'] / (total_pp['Miskin'] + total_pp['Tidak Miskin'])
            self.prior_probability['Tidak Miskin'] = total_pp['Tidak Miskin'] / (total_pp['Miskin'] + total_pp['Tidak Miskin'])
            # TODO: [SOAL-2] Prior Probability-nya masih 0, hitunglah prior probability yang sebenarnya!

        # TODO: [LANGKAH-9] Membuat method untuk memprediksi hasil akhir berdasarkan nilai aspek
        def prediksi(self, nilai_umur: str, nilai_status: str, nilai_pendidikan: str, nilai_tanggungan: str, nilai_pekerjaan: str, nilai_penghasilan: str):

            self.hitung_prior_probability()
            predict_miskin = self.prior_probability['Miskin'] * \
                self.aspek_umur[nilai_umur].p_aspek_miskin * \
                self.aspek_status[nilai_status].p_aspek_miskin * \
                self.aspek_pendidikan[nilai_pendidikan].p_aspek_miskin * \
                self.aspek_tanggungan[nilai_tanggungan].p_aspek_miskin * \
                self.aspek_pekerjaan[nilai_pekerjaan].p_aspek_miskin * \
                self.aspek_penghasilan[nilai_penghasilan].p_aspek_miskin
            print('Peluang Miskin: {}'.format(predict_miskin))

            predict_tidak_miskin = self.prior_probability['Tidak Miskin'] * \
                self.aspek_umur[nilai_umur].p_aspek_tidak_miskin * \
                self.aspek_status[nilai_status].p_aspek_tidak_miskin * \

```

```

        self.aspek_pendidikan[nilai_pendidikan].p_aspek_tidak_miskin * \
        self.aspek_tanggungan[nilai_tanggungan].p_aspek_tidak_miskin * \
        self.aspek_pekerjaan[nilai_pekerjaan].p_aspek_tidak_miskin * \
        self.aspek_penghasilan[nilai_penghasilan].p_aspek_tidak_miskin

    print('Peluang Tidak Miskin: {}'.format(predict_tidak_miskin))

    if predict_tidak_miskin > predict_miskin:
        hasil = "Tidak Miskin"
        peluang = predict_tidak_miskin
    else:
        hasil = "Miskin"
        peluang = predict_miskin
    return {'hasil': hasil, 'peluang': peluang}

```

Membuat ProbAspek.py

```

class ProbAspek:

    def __init__(self, nama_aspek: str, nilai_aspek: str):
        pass

    # TODO: [LANGKAH-1] Buat class untuk menampung nilai matriks probabilitas

    self.nama_aspek = nama_aspek
    self.nilai_aspek = nilai_aspek
    self.jml_miskin = 0
    self.jml_tidak_miskin = 0
    self.p_aspek_miskin = 0
    self.p_aspek_tidak_miskin = 0

    #
    def hitung_p_aspek_miskin(self, jml_total_miskin_aspek):
        # try:
        self.p_aspek_miskin = self.jml_miskin / jml_total_miskin_aspek
        # except ZeroDivisionError:
        #     self.p_aspek_miskin = 0

        return self

    #
    def hitung_p_aspek_tidak_miskin(self, jml_total_tidak_miskin_aspek):
        # try:

```



```

        self.p_aspek_tidak_miskin = self.jml_tidak_miskin / jml_total_tidak_miskin
n_aspek
    # except ZeroDivisionError:
    #     self.p_aspek_tidak_miskin = 0

    return self

#
def print(self):
    print('Aspek      : {}'.format(self.nama_aspek))
    print('Nilai      : {}'.format(self.nilai_aspek))
    print('Jml Miskin: {}'.format(self.jml_miskin))
    print('Jml Tidak Miskin: {}'.format(self.jml_tidak_miskin))
    print('P({}|Miskin): {}'.format(self.nilai_aspek, self.p_aspek_miskin))
    print('P({}|Tidak Miskin): {}'.format(self.nilai_aspek, self.p_aspek_tidak_miskin))
    print('-----')

#
@staticmethod
def hitung_jml_total_aspek(pa_list: list) -> dict:
    jumlah = {'Miskin': 0, 'Tidak Miskin': 0}
    for pa in pa_list:
        jumlah['Miskin'] += pa.jml_miskin
        jumlah['Tidak Miskin'] += pa.jml_tidak_miskin
    return jumlah

#
@staticmethod
def print_matrix_probabilitas(pa_list: list):
    for pa in pa_list:
        pa.print()

```

## Main.py

```

from naivebayes import NaiveBayes

class Main:
    @staticmethod
    def main():
        nb = NaiveBayes()
        nb.load_data_training()
        nb.mulai_training()
        nilaiUmur = input("Masukkan nilai Umur[Tua/Muda] : ")
        nilaiStatus = input("Masukkan nilai Status [Belum Kawin/Kawin/Cerai]: ")

```

```

        nilaiPendidikan = input("Masukkan nilai Pendidikan [Tidak Sekolah/SD/SLTP
/SLTA/Sarjana] : ")
        nilaiTanggungan = input("Masukkan nilai Tanggungan [Tidak Ada/1/2/3/4/5]:
")
        nilaiPekerjaan = input("Masukkan nilai Pekerjaan [Tiada/Buruh Lepas/Petan
i/Aparatur Negara] : ")
        nilaiPenghasilan = input("Masukkan nilai Penghasilan [Tiada/Rendah/Sedang
/Tinggi] : ")

        hasil_prediksi = nb.prediksi(nilai_umur=nilaiUmur,
                                       nilai_status=nilaiStatus,
                                       nilai_pendidikan=nilaiPendidikan,
                                       nilai_tanggungan=nilaiTanggungan,
                                       nilai_pekerjaan=nilaiPekerjaan,
                                       nilai_penghasilan=nilaiPenghasilan)

        print('=====')

        print('Hasil akhir prediksi = {}, dengan peluang sebesar {}'.format(hasi
l_prediksi['hasil'],
                                                                    hasil
l_prediksi['peluang'])))

Main.main()

```

Bikin data daftar masyarakat.csv

daftar\_masyarakat\_miskin.csv - Excel (Product Activation Failed)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	id;Umur;Status;Pendidikan;Tanggungan;Pekerjaan;Penghasilan;Actual Class																
2	1	Tua	Kawin	SLTP	1	Pedagang	Tinggi	Tidak Miskin									
3	2	Muda	Belum Kawin	SD	Tidak Ada	Buruh Lepas	Rendah	Miskin									
4	3	Tua	Kawin	Sarjana	3	Aparatur Negara	Tinggi	Tidak Miskin									
5	4	Tua	Kawin	SLTP	4	Petani	Rendah	Miskin									
6	5	Muda	Kawin	Sarjana	4	Aparatur Negara	Tinggi	Tidak Miskin									
7	6	Muda	Kawin	SD	3	Petani	Rendah	Miskin									
8	7	Muda	Kawin	Tidak Sekolah	2	Petani	Sedang	Miskin									
9	8	Tua	Kawin	SLTP	3	Buruh Lepas	Rendah	Miskin									
10	9	Tua	Cerai	SLTA	2	Tiada	Tiada	Miskin									
11	10	Muda	Kawin	SD	3	Petani	Rendah	Miskin									
12	11	Muda	Kawin	SLTP	3	Petani	Sedang	Tidak Miskin									
13	12	Muda	Belum Kawin	SD	Tidak Ada	Buruh Lepas	Rendah	Tidak Miskin									
14	13	Tua	Kawin	SD	1	Tiada	Tiada	Miskin									
15	14	Muda	Kawin	SD	3	Buruh Lepas	Rendah	Miskin									
16	15	Muda	Kawin	SD	5	Petani	Sedang	Miskin									
17	16	Tua	Kawin	SD	3	Petani	Rendah	Miskin									
18	17	Tua	Cerai	SLTA	1	Tiada	Tiada	Miskin									
19	18	Tua	Cerai	SD	2	Tiada	Tiada	Miskin									
20	19	Tua	Kawin	SD	3	Buruh Lepas	Rendah	Miskin									
21																	
22																	

Buat data daftar\_masyarakat\_miskin.xlsx

daftar\_masyarakat\_miskin.xlsx [Protected View] - Excel (Product Activation Failed)

**PROTECTED VIEW** Be careful—files from the Internet can contain viruses. Unless you need to edit, it's safer to stay in Protected View. [Enable Editing](#)

	A	B	C	D	E	F	G	H	I	J	K	L	M
10	9	Tua	Cerai	SLTA	2	Tiada	Tiada	MISKIN					
11	10	Muda	Kawin	SD	3	Petani	Rendah	MISKIN					
12	11	Muda	Kawin	SLTP	3	Petani	Sedang	TIDAK MISKIN					
13	12	Muda	Belum Kawin	SD	Tidak Ada	Buruh Lepas	Rendah	TIDAK MISKIN					
14	13	Tua	Kawin	SD	1	Tiada	Tiada	MISKIN					
15	14	Muda	Kawin	SD	3	Buruh Lepas	Rendah	MISKIN					
16	15	Muda	Kawin	SD	5	Petani	Sedang	MISKIN					
17	16	Tua	Kawin	SD	3	Petani	Rendah	MISKIN					
18	17	Tua	Cerai	SLTA	1	Tiada	Tiada	MISKIN					
19	18	Tua	Cerai	SD	2	Tiada	Tiada	MISKIN					
20	19	Tua	Kawin	SD	3	Buruh Lepas	Rendah	MISKIN					
21													
22													
23													

Kita run main.py

Muncul data

```

/Local/Programs/Python/Python39/python.exe" "d:/TUGAS SEKOLAH BAGUS SATHIA/KULIAH/Semester 6/Teknologi Data/UAS/Klasifikasi_Masyarakat_Miskin (1)/Klasifikasi_Masyara
kat_Miskin/main.py"
& "C:/Users/ROG SERIES/AppData

```

id	Umur	Status	Pendidikan	Tanggungan	Pekerjaan	Penghasilan	Actual Class
0	1	Tua	Kawin	SLTP	1	Pedagang	Tinggi Tidak Miskin
1	2	Muda	Belum Kawin	SD	Tidak Ada	Buruh Lepas	Rendah Miskin
2	3	Tua	Kawin	Sarjana	3	Aparatur Negara	Tinggi Tidak Miskin
3	4	Tua	Kawin	SLTP	4	Petani	Rendah Miskin
4	5	Muda	Kawin	Sarjana	4	Aparatur Negara	Tinggi Tidak Miskin
5	6	Muda	Kawin	SD	3	Petani	Rendah Miskin
6	7	Muda	Kawin	Tidak Sekolah	2	Petani	Sedang Miskin
7	8	Tua	Kawin	SLTP	3	Buruh Lepas	Rendah Miskin
8	9	Tua	Cerai	SLTA	2	Tiada	Miskin
9	10	Muda	Kawin	SD	3	Petani	Rendah Miskin
10	11	Muda	Kawin	SLTP	3	Petani	Sedang Tidak Miskin
11	12	Muda	Belum Kawin	SD	Tidak Ada	Buruh Lepas	Rendah Tidak Miskin
12	13	Tua	Kawin	SD	1	Tiada	Tiada Miskin
13	14	Muda	Kawin	SD	3	Buruh Lepas	Rendah Miskin
14	15	Muda	Kawin	SD	5	Petani	Sedang Miskin
15	16	Tua	Kawin	SD	3	Petani	Rendah Miskin
16	17	Tua	Cerai	SLTA	1	Tiada	Tiada Miskin
17	18	Tua	Cerai	SD	2	Tiada	Tiada Miskin
18	19	Tua	Kawin	SD	3	Buruh Lepas	Rendah Miskin

Muncul data yang diurutkan secara vertical

```

-----
Aspek    : Umur
Nilai    : Tua
Jml Miskin: 8
Jml Tidak Miskin: 2
P(Tua|Miskin): 0.5714285714285714
P(Tua|Tidak Miskin): 0.4
-----

Aspek    : Umur
Nilai    : Muda
Jml Miskin: 6
Jml Tidak Miskin: 3
P(Muda|Miskin): 0.42857142857142855
P(Muda|Tidak Miskin): 0.6
-----

Aspek    : Status
Nilai    : Belum Kawin
Jml Miskin: 1
Jml Tidak Miskin: 1
P(Belum Kawin|Miskin): 0.07142857142857142
P(Belum Kawin|Tidak Miskin): 0.2
-----

Aspek    : Status
Nilai    : Kawin
Jml Miskin: 10
Jml Tidak Miskin: 4
P(Kawin|Miskin): 0.7142857142857143
P(Kawin|Tidak Miskin): 0.8
-----

Aspek    : Status
Nilai    : Cerai
Jml Miskin: 3
Jml Tidak Miskin: 0
P(Cerai|Miskin): 0.21428571428571427
P(Cerai|Tidak Miskin): 0.0
-----

Aspek    : Pendidikan
Nilai    : Tidak Sekolah

```

Jml Miskin: 1  
Jml Tidak Miskin: 0  
P(Tidak Sekolah|Miskin): 0.07142857142857142  
P(Tidak Sekolah|Tidak Miskin): 0.0

-----  
Aspek : Pendidikan  
Nilai : SD  
Jml Miskin: 9  
Jml Tidak Miskin: 1  
P(SD|Miskin): 0.6428571428571429  
P(SD|Tidak Miskin): 0.2

-----  
Aspek : Pendidikan  
Nilai : SLTP  
Jml Miskin: 2  
Jml Tidak Miskin: 2  
P(SLTP|Miskin): 0.14285714285714285  
P(SLTP|Tidak Miskin): 0.4

-----  
Aspek : Pendidikan  
Nilai : SLTA  
Jml Miskin: 2  
Jml Tidak Miskin: 0  
P(SLTA|Miskin): 0.14285714285714285  
P(SLTA|Tidak Miskin): 0.0

-----  
Aspek : Pendidikan  
Nilai : Sarjana  
Jml Miskin: 0  
Jml Tidak Miskin: 2  
P(Sarjana|Miskin): 0.0  
P(Sarjana|Tidak Miskin): 0.4

-----  
Aspek : Tanggungan  
Nilai : Tidak Ada  
Jml Miskin: 1  
Jml Tidak Miskin: 1  
P(Tidak Ada|Miskin): 0.07142857142857142  
P(Tidak Ada|Tidak Miskin): 0.2

-----  
Aspek : Tanggungan

Nilai : 1  
Jml Miskin: 2  
Jml Tidak Miskin: 1  
P(1|Miskin): 0.14285714285714285  
P(1|Tidak Miskin): 0.2

-----  
Aspek : Tanggungan  
Nilai : 2  
Jml Miskin: 3  
Jml Tidak Miskin: 0  
P(2|Miskin): 0.21428571428571427  
P(2|Tidak Miskin): 0.0

-----  
Aspek : Tanggungan  
Nilai : 3  
Jml Miskin: 6  
Jml Tidak Miskin: 2  
P(3|Miskin): 0.42857142857142855  
P(3|Tidak Miskin): 0.4

-----  
Aspek : Tanggungan  
Nilai : 4  
Jml Miskin: 1  
Jml Tidak Miskin: 1  
P(4|Miskin): 0.07142857142857142  
P(4|Tidak Miskin): 0.2

-----  
Aspek : Tanggungan  
Nilai : 5  
Jml Miskin: 1  
Jml Tidak Miskin: 0  
P(5|Miskin): 0.07142857142857142  
P(5|Tidak Miskin): 0.0

-----  
Aspek : Pekerjaan  
Nilai : Tiada  
Jml Miskin: 4  
Jml Tidak Miskin: 0  
P(Tiada|Miskin): 0.2857142857142857  
P(Tiada|Tidak Miskin): 0.0  
-----

P(Tiada|Tidak Miskin): 0.0

-----  
Aspek : Pekerjaan  
Nilai : Buruh Lepas  
Jml Miskin: 4

Jml Tidak Miskin: 1  
P(Buruh Lepas|Miskin): 0.2857142857142857  
P(Buruh Lepas|Tidak Miskin): 0.25

-----  
Aspek : Pekerjaan  
Nilai : Petani  
Jml Miskin: 6

Jml Tidak Miskin: 1  
P(Petani|Miskin): 0.42857142857142855  
P(Petani|Tidak Miskin): 0.25

-----  
Aspek : Pekerjaan  
Nilai : Aparatur Negara  
Jml Miskin: 0

Jml Tidak Miskin: 2  
P(Aparatur Negara|Miskin): 0.0  
P(Aparatur Negara|Tidak Miskin): 0.5

-----  
Aspek : Penghasilan  
Nilai : Tiada  
Jml Miskin: 4

Jml Tidak Miskin: 0  
P(Tiada|Miskin): 0.2857142857142857  
P(Tiada|Tidak Miskin): 0.0

-----  
Aspek : Penghasilan  
Nilai : Rendah  
Jml Miskin: 8

Jml Tidak Miskin: 1  
P(Rendah|Miskin): 0.5714285714285714  
P(Rendah|Tidak Miskin): 0.2

-----  
Aspek : Penghasilan  
Nilai : Sedang  
Jml Miskin: 2

Jml Tidak Miskin: 1

	<pre> ----- Aspek      : Penghasilan Nilai      : Tinggi Jml Miskin: 0 Jml Tidak Miskin: 3 P(Tinggi Miskin): 0.0 P(Tinggi Tidak Miskin): 0.6 ----- </pre>
	<p>Kita input ke pemrograman tersebut</p> <pre> Masukkan nilai Umur[Tua/Muda] : Muda Masukkan nilai Status [Belum Kawin/Kawin/Cerai]: Belum Kawin Masukkan nilai Pendidikan [Tidak Sekolah/SD/SLTP/SLTA/Sarjana] : SLTA Masukkan nilai Tanggungan [Tidak Ada/1/2/3/4/5]: 3 Masukkan nilai Pekerjaan [Tiada/Buruh Lepas/Petani/Aparatur Negara] : Petani Masukkan nilai Penghasilan [Tiada/Rendah/Sedang/Tinggi] : Sedang Peluang Miskin: 8.455123648354536e-05 Peluang Tidak Miskin: 0.0 ===== </pre>
	<p>Muncul Hasil akhir sebagai berikut</p> <pre> Hasil akhir prediksi = Miskin, dengan peluang sebesar 8.455123648354536e-05% </pre>

**--Alhamdulillah--**